HANDWRITING TEXT INPUT SYSTEM

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates to improvements of a handwriting text input system comprising an input system, a memory system, a dictionary system, an output system (a display system) including a program to output on the display the input data, and to edit data in the text, to do a dictionary search for the spelling check of input data and conversion of an abbreviation input data to corresponding words in the memory.

2. Description of the Prior Art

The conventional system of text input system has a way of using an abbreviation (e.g. assoc for association, govt for government, etc) for a word being frequently used, as well as a ~~shorhand~~ shorthand (e.g. e for editing, l for loading, etc) that assigns a single character to a longer line of text or commands, to lower the burden of the text input.

However, it is inconvenient ~~in~~ that ~~such way should get~~ a person must remember each of the abbreviation and shorthand which must be typed as was previously assigned and must be actuated by the corresponding key. Accordingly, the ~~it's higher~~ input performance depends on each person's capability.

SUMMARY OF THE INVENTION

The present invention was made to improve such defect of conventional system, providing a natural and efficient way of text input without particular skills ~~to use it and without being aware of the invention method in use.~~

According to the present invention there is provided a text input system comprising;

input system for inputting a plurality of word data;

dictionary system for storing a plurality of a line of text or pattern element data to represent a unique original word in said dictionary, original word data and relevant word data for said original word data;

system for determining said unique line of text or pattern element data which represents said unique word and which includes said data word from said input system, in said dictionary system;

system for selecting a desired word among said relevant words where ~~in case of that~~ said unique word has said relevant words in said dictionary, after successful execution of said system for determining said unique line of text or pattern element data;

output system for outputting said unique word represented by said line of text or pattern element data which was determined by said system for determining, and outputting said unique word selected by said system for selecting.

The present invention has a dictionary to find a unique word data in comparison with the data input, character by character, or (pattern element) data by (pattern element) data, and automatically supplements the

remaining part of the data from the dictionary, if the dictionary has the word which is unique and which includes word data in the input buffer.

Some examples for the number of character positions in a word to be unique in the dictionary in comparison with the total number of characters of the word, are given below. The word in the examples could be a line of text or word pattern element data to represent an original word data.

Also, there is a possibility to be more quick in finding a unique word in the dictionary, by decreasing the number of short words in the dictionary. For example, "ability" can be identified at its 3rd character positions (i.e., abi), if "abide" is not in the following dictionary.

Word (or line of text)	The number of character position in a words to be unique in the dictionary	The total number of characters of the word
abandon	4	7
abate	4	5
abbot	4	5
abdomen	3	7
abhor	3	5
abide	4	5
ability	4	7
abject	3	6
able	4	4
abnormal	3	8
aboard	4	6
abolish	6	7
abolition	6	9

abominable	4	10
abound	5	6
about	5	5
above	4	5
abridge	4	7
abroad	4	6
abrupt	4	6
absence	6	7
absent	6	6
absinthe	4	8
absolute	6	8
absolve	6	7
absorb	5	6
absorbent	7	9
abstain	5	7
(Total)	127	184

127 / 184 = 0.69 (31% difference)

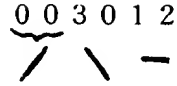
Handwriting character pattern element codes — original word

02011211102 — 正

02112512120030 — 更

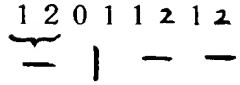
In this example, "正" or "更" can be unique at the second pattern element code position.

003012 — A



0134 — B

12011212 — E



BRIEF DESCRIPTION OF THE DRAWINGS

~~By way of example, the system of the present invention is described in a flowchart in Fig. 2 to Fig. 10 and Fig. 11 to Fig. 14 show examples of the dictionary to use by the invention system. Fig. 1 is a block diagram of the invention system.~~

Ins 9

DETAILED DESCRIPTION OF THE DRAWINGS

In the flowchart, A001 is the first step to clear the contents of the program counter, flag, and the working area.

A002 accepts data from input system, character by character, or (pattern element) data by (pattern element) data.

A003 checks if the input is the end of data code, and goes to the stop point of this process if it is the end. Otherwise, the step proceeds to the next A003A.

A003A tests if the NONE FLAG is ON which means that there is no word including word data input, in the dictionary.

If ON, the step goes to S020.

If OFF, the step proceeds to A003B.

A003B tests if the input is a separator code. If it is the separator, A003C takes place to clear the contents of input buffer because of the termination of the word input and then the step goes back to A002 for the next data input.

A004 arises if the input is not the separator code at A003B and the input data

is displayed at the end of data on the display.

Next A005 is the step to add the input data to the end of previous data in the input buffer memory.

A006 performs a dictionary search to know if the input data is included in the line of text or pattern element data to represent unique word in the dictionary. (1)

A006A tests the result of the dictionary search at A006.

In case of negative result, A006B takes place to turn on the none flag and go back to A002. Affirmative result brings the process to the next A007.

A007 confirms if A006 found a single one by testing the flag for the unique word. If not, the step goes back to A002. Otherwise, A008 takes place in which the input data on the display is replaced with those of the unique data from the dictionary and contents of the input buffer is cleared. (2)

At next, A008A, the selection of a desired word takes place by system for selecting the desired data out of relevant data in case of having relevant data in the dictionary, after successful execution of system for determining unique line of text or pattern element data, by the selection of the desired data which is unique and terminates with the same one as the last input of word data from input system, or which is unique and includes the same one as the last input of word data from input system, in the remaining positions of relevant words other than what was collated, in said unique line of text or pattern element data determined. Then, jumps back to A002.

S001 is an entry of the subprogram to search the unique line of text or pattern element data to represent the unique word in the dictinary.

S002 is to test if there is no

more data to retrieve in the dictionary. If not so, the process goes to the next S003. Otherwise, it goes to the exit of this subprogram.

S003 gets the data located in the middle of the area between upper limit and lower limit in the dictionary.

Next S004 checks if the leading part of the data from dictionary is greater than that of word data input.

If the greater flag is ON at S004, S005 gets up to divide the retrieval area size into 2 to use the lower half area for the next retrieval, as there is a possibility to find the unique data in the lower half area, because the dictionary data found at S003 was greater than the data input. Then, the step goes back to S002. If the greater than flag is OFF at S004, the step proceeds to S006.

At S006, it is tested if the leading part of the data from the dictionary is equal to that of the data input.

If equal, S007 is performed to test if it is single.

In case of a single one, next step S008 turns on the flag for the unique data in the dictionary and goes to exit.

If there are plural data, the step goes to S012.

If not equal at S006, the step proceeds to S009 in which the testing is made if the leading part of the data from the dictionary is less than that of the data input.

If the less than flag is ON after testing at S009, S010 divides the retrieval area size into 2 to use the upper half area for the next retrieval, as there is a possibility to find the unique data in the upper half area because the dictionary data found at S003 was less than that of data input.

If the less than flag is OFF at S009, there is something wrong with the

procedure in the program and goes to the error procedure of S011 which is not described here.

Jumping here from S007, S012 tests if plural data found at S007 have the same stem of a word. If YES, S012A turns on the NONE FLAG and goes back to A002. This is to use the ID CHECK subprogram. If NO, the step goes to S012B.

S012B tries to find the data with the leading part greater than that of the data input in the upper half of the retrieval area in the dictionary.

At the next S013, the step goes to S014 if no data was found, and goes to S015 if found.

S014 sets new upper limit to retrieve in the dictionary using the position of the upper limit by which S012 was attempted because the upper half area in this case is occupied by the data with the leading part equal to that of the data input. Then, the step goes to S016.

S015 sets new upper limit to retrieve in the dictionary using the position of the data found at the step of S012.

S016 tries to find the data with the leading part less than that of the data input in the lower half area of the dictionary.

S017 tests if the data was found.

S018 takes place if not found at S017 and sets new lower limit using the position of the lower limit by which S016 was attempted because the lower area is occupied by the data with the leading part equal to that of the data input. Then, the step goes to the exit of this subprogram.

S019 arises if found at S017 and sets new lower limit to retrieve using the position of the data with the leading part less than that of the data input and jumps to the exit of this subprogram.

not clear

Branching from A003A when the NONE FLAG was on, S020 performs ID CHECK to start with S023 that is an entry of subprogram to test if there is a unique data which terminates with the same one as the last input data, or which includes the same one as the last input data in the remaining positions other than what was collated with the data input.

Next S021 checks the FLAG FOR UNIQUE WORD after coming back from ID CHECK.

If it is OFF, the step goes back to A002.

If it is ON, the step proceeds to the next S022 which turns off the NONE FLAG and supplements the remainder of data input on the display.

Then, the step goes back to A002.

S023 is the start of subprogram which performs determination of unique data.

S023 examines data which were selected by the prior lookup in the dictionary.

S024 tests if there is unique data which terminates with the same one as the last input. If NO, the step goes to S026.

If YES, the step proceeds to the next S025 which turns on the FLAG FOR UNIQUE WORD and goes to exit of this subprogram.

At S026, it is tested if there is unique data which includes the last input in the remaining positions other than what was collated with the data input.

If YES, the step goes to S025.

If NO, the step proceeds to the next S027 in which the system extracts data which does not terminate with the same one as the last input or which does not include the same one as the last input in the remaining positions other than what was collated with the data input.

As a subset of a dictionary search for the word data input consisting of a first character and some other characters, T001 which is equivalent to the aforementioned S001, is an entry of this subprogram.

T002 which is equivalent to the aforementioned S002 checks if there is still more word to retrieve in the dictionary. If there is no more word, the step goes to the exit. The process advances to the next T002A, if there are some more words to retrieve.

T002A branches to the aforementioned S003 if the input is the first one of the word data input, and to the next T002B if it was not first.

T002B is to check if all words between upper and lower limit in the dictionary have the same leading part as that of the word data input.

T002C selects the words with the leading part which is not same as that of the word data input.

T002D shifts one character to the left for every words selected at T002C and extract those words if terminated after making a character shift.

T002E sets new upper and lower limit to retrieve in the dictionary after extracing words terminated at T002D. Then, the step goes to the aforementioned S003.